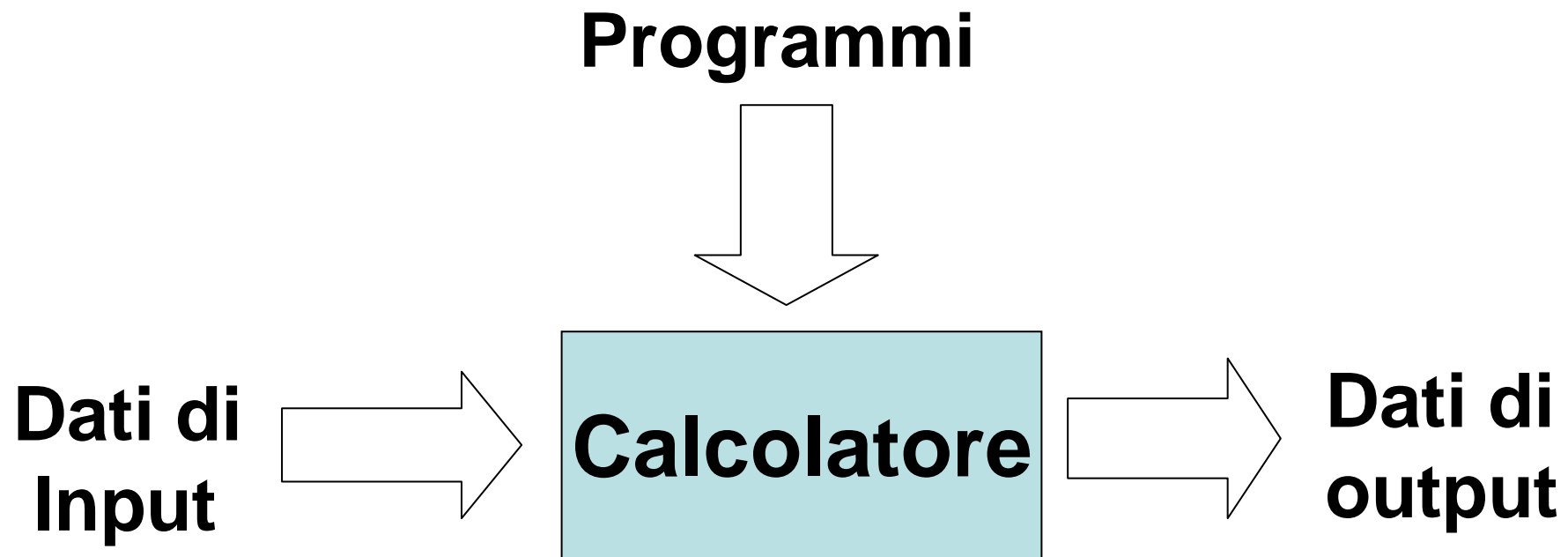


Architettura del Calcolatore

Modulo 1

Schema di un Calcolatore



macchina  **hardware**

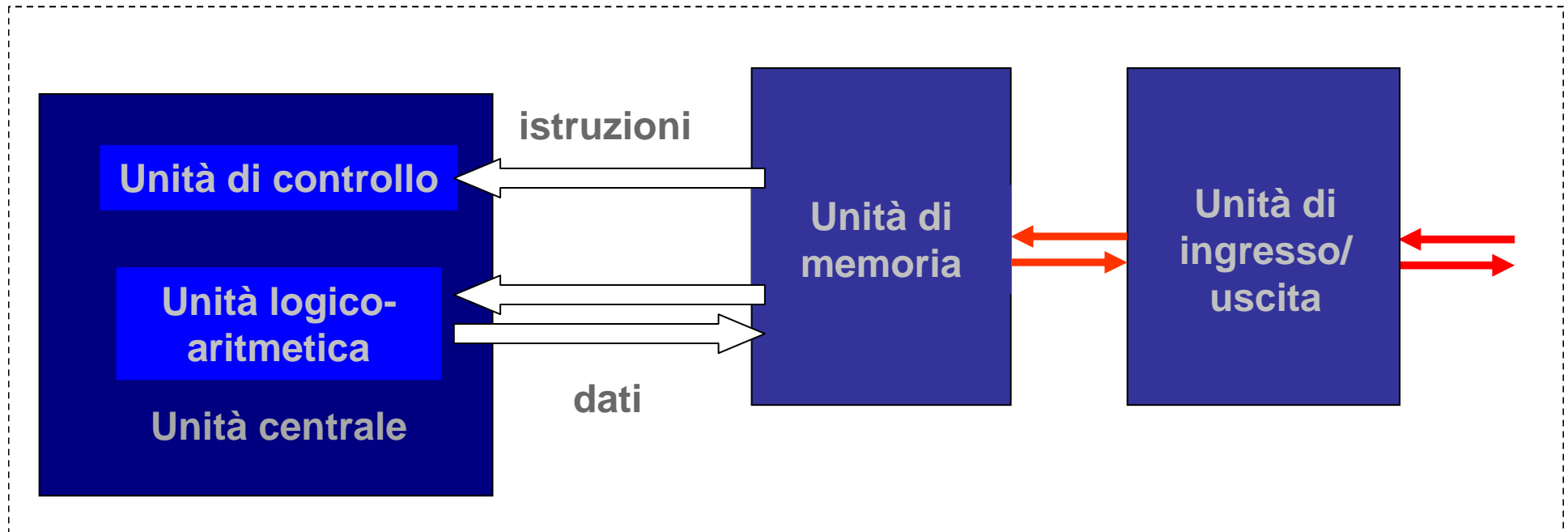
struttura fisica del calcolatore, definita dall'insieme delle unità funzionali che la compongono e dalle loro interconnessioni

programma  **software**

insieme di istruzioni da eseguire secondo un ordine preciso, il cui effetto è la realizzazione di uno specifico compito

calcolatore = hardware + software

Modello di von Neumann

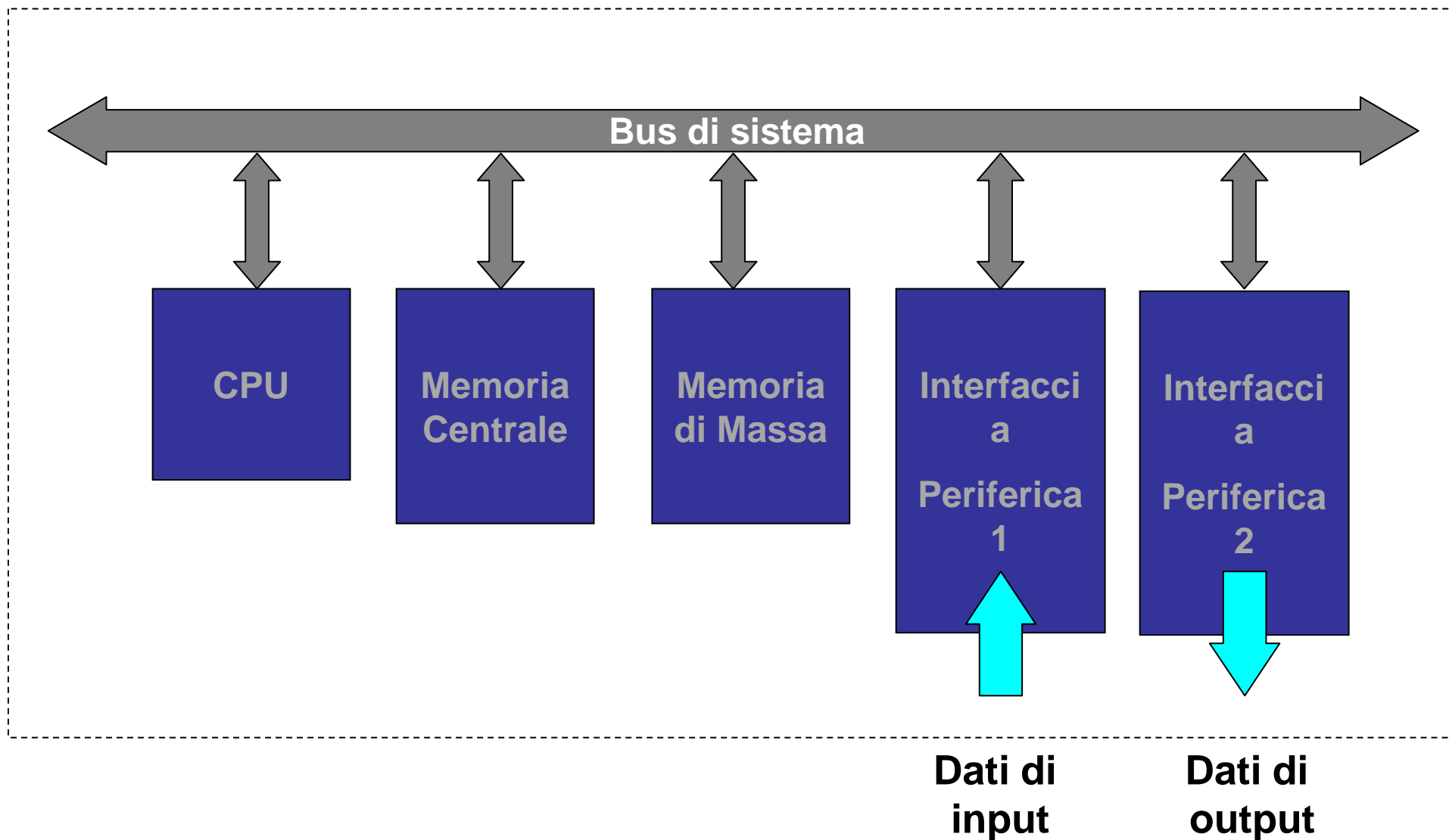


Modello logico

single componenti

flussi di dati e istruzioni

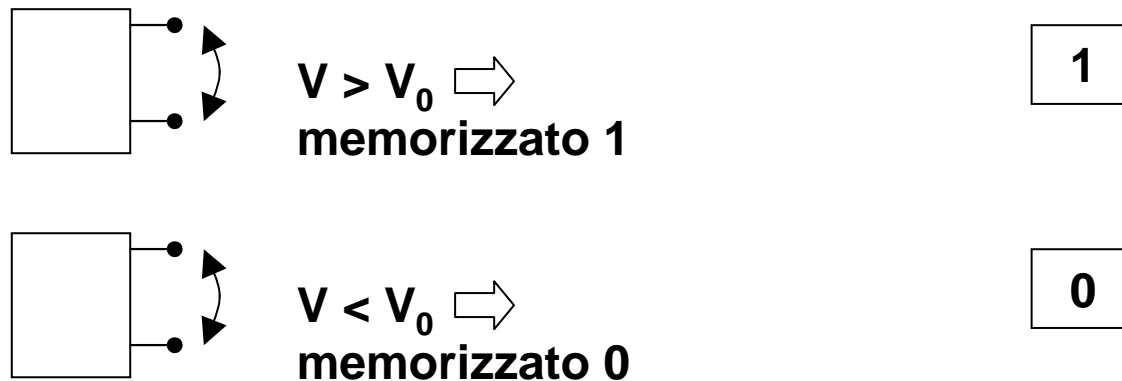
Un implementazione del modello di Von Neumann



La memorizzazione dei dati e delle istruzioni

La più piccola unità di informazione memorizzabile (e quindi utilizzabile) è il **bit (Binary digIT)**, che può assumere valore 0 o 1.

Il dispositivo utilizzato per memorizzare un bit è un **elemento bistabile**, cioè un dispositivo elettronico che può assumere uno tra due stati stabili (es. due livelli differenti di tensione), ognuno dei quali viene fatto corrispondere a 0 o a 1 (cella di memoria).



Operazioni possibili su una cella di memoria

Operazione di scrittura

La cella di memoria viene caricata con un determinato valore che permane memorizzato finchè:

- la cella viene alimentata elettricamente
- non si esegue un'altra operazione di scrittura che modifica il valore precedentemente memorizzato

Operazione di lettura

Si accede alla cella di memoria per consultarne il valore e copiarlo su un'altra cella di memoria, senza alterarne il contenuto

Nota

Non su tutte le celle di memoria sono possibili entrambe le operazioni di lettura e scrittura.

Con un solo bit è possibile gestire un'informazione binaria, cioè un'informazione che può specificare uno tra due valori possibili (es. un punto di un'immagine bianco o nero).

Quanti stati possibili può assumere un insieme di bit ?

00	000	0000	
01	001	0001	
10	010	0010	2 bit → 4 stati
11	011	0011	3 bit → 8 stati
	100	0100	4 bit → 16 stati
	101	0101	...
	110	0110	
	111	0111	
		1000	
		1001	
		1010	
		1011	
		1100	
		1101	
		1110	
		1111	

Il registro di memoria

Un insieme di N celle elementari può assumere uno tra 2^N stati possibili.

Un tale insieme è organizzato in un **registro di memoria**.

Il registro costituisce un supporto per la memorizzazione di un'informazione che può assumere uno tra 2^N valori possibili. In particolare un insieme di 8 bit forma un **byte**.

Sul registro sono possibili operazioni di lettura e scrittura che interessano contemporaneamente tutte le celle di memoria contenute nel registro.

Il problema della codifica

Un calcolatore può trattare diversi tipi di dati: numeri (interi, reali), testo, immagini, suoni, ecc. che vanno comunque memorizzati su registri di memoria.

È quindi necessario adottare una **codifica** del tipo di dato considerato: occorre, cioè,

mettere in corrispondenza biunivoca i valori del tipo con gli stati che può assumere il registro.

Esempio

registro da un byte $\Rightarrow 2^8 = 256$ stati possibili.

Che cosa è possibile codificare ?

Numeri naturali [0,255]

0 \leftrightarrow 00000000

1 \leftrightarrow 00000001

....

255 \leftrightarrow 11111111

Numeri interi [-128,127]

-128 \leftrightarrow 00000000

-127 \leftrightarrow 00000001

0 \leftrightarrow 10000000

+127 \leftrightarrow 11111111

Numeri reali [0,1[

0.0000 \leftrightarrow 00000000

0.0039 \leftrightarrow 00000001

0.0078 \leftrightarrow 00000010

....

0.9961 \leftrightarrow 11111111

Caratteri

A \leftrightarrow 01000001

a \leftrightarrow 01100001

0 \leftrightarrow 00110000

1 \leftrightarrow 00110001

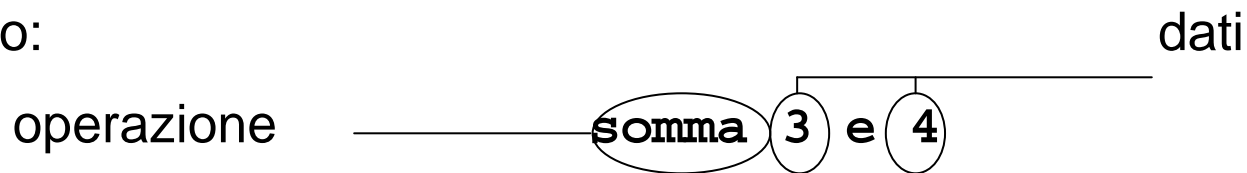
La codifica implica una rappresentazione dei dati limitata e discreta

Codifica delle istruzioni

Oltre ai dati, è necessario memorizzare anche le istruzioni, cioè le singole azioni elementari che l'unità centrale può eseguire.

Nello specificare un'istruzione, bisogna precisare l'operazione da compiere e i dati coinvolti nell'operazione.

Esempio:



Come rappresentare le operazioni ?

L'insieme delle diverse operazioni che l'unità centrale è in grado di eseguire è finito e quindi è possibile codificarlo con un certo numero di bit (**codice operativo**).

somma	0000
sottrai	0001
moltiplica	0010
dividi	0011
...	...

Una istruzione sarà quindi rappresentabile da una sequenza di bit divisa in due parti:

- un codice operativo
- un campo operandi (1, 2 o più operandi)



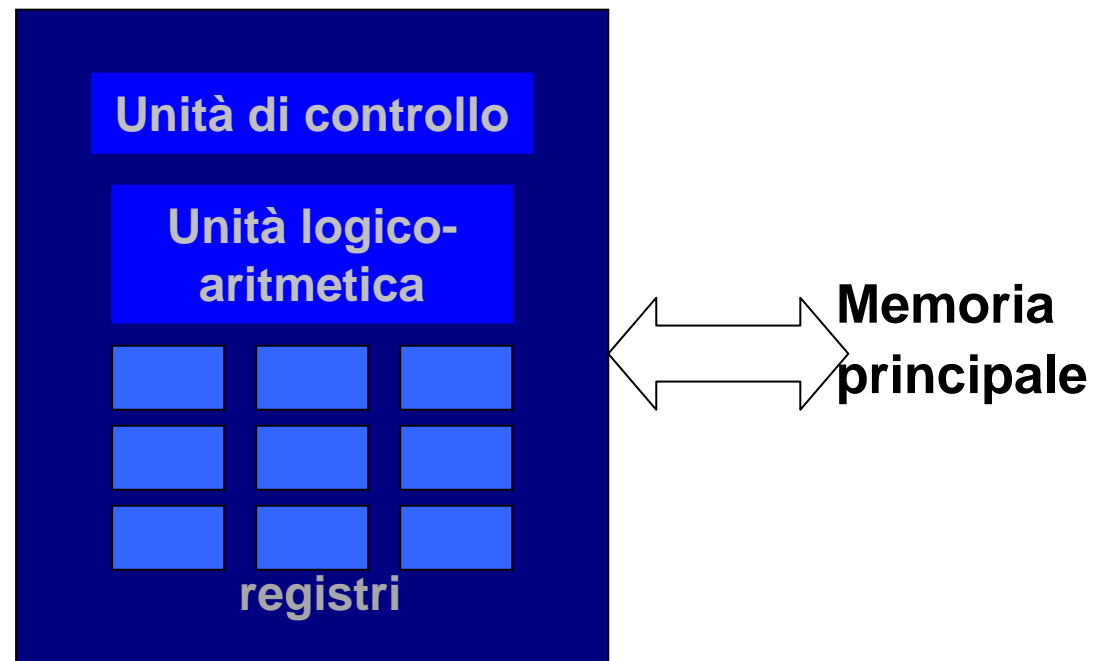
CPU (Central Processing Unit)

Funzione:

eseguire i programmi immagazzinati in memoria principale prelevando le istruzioni (e i dati relativi), interpretandole ed eseguendole una dopo l'altra

E' formata da:

- **unità di controllo**
- **unità logico aritmetica**
- **registri**



L'Unità di controllo (1/2)

E' l'unità che si occupa di dirigere e coordinare le attività interne alla CPU che portano all'esecuzione di una istruzione

Ciclo del processore

L'esecuzione di una istruzione avviene attraverso alcune fasi:

Fetch

L'istruzione da eseguire viene prelevata dalla memoria e trasferita all'interno della CPU

Decode

L'istruzione viene interpretata e vengono avviate le azioni interne necessarie per la sua esecuzione

Operand Assembly

Vengono prelevati dalla memoria i dati su cui eseguire l'operazione prevista dalla istruzione

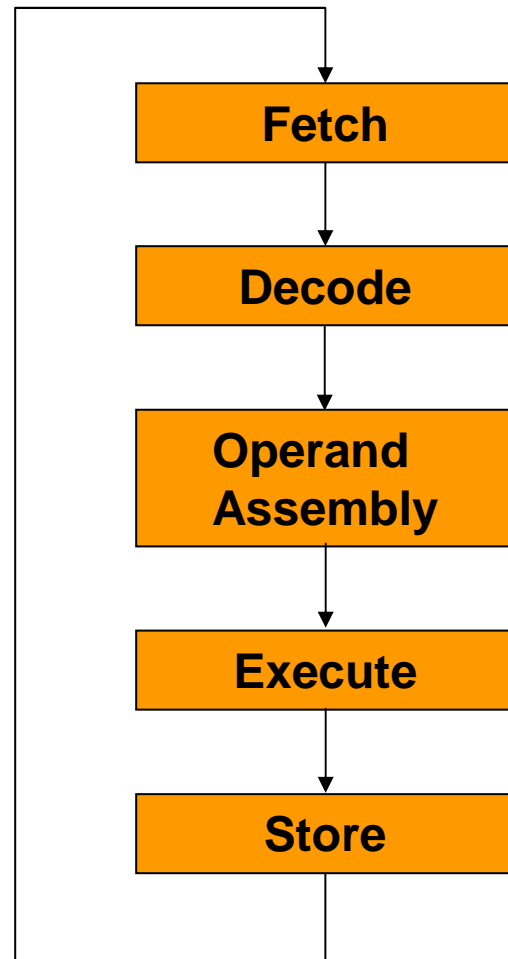
Execute

Viene portata a termine l'esecuzione dell'operazione prevista dalla istruzione

Store

Viene memorizzato il risultato dell'operazione prevista dalla istruzione

L'Unità di controllo (2/2)



L'unità di controllo realizza in ciclo le fasi per eseguire la sequenza di istruzioni che costituiscono il programma

L'Unità Logico Aritmetica

E' l'unità che si occupa di realizzare le operazioni logiche ed aritmetiche eventualmente richieste per eseguire un'istruzione

Operazioni Aritmetiche

ADD

SUB

MUL

DIV

REM

SET

Operazioni Logiche

CMP

AND

OR

NOT

I registri

Hanno la funzione di memorizzare all'interno della CPU dati e istruzioni necessari all'esecuzione

- **Registri generali**

- **Registri speciali**

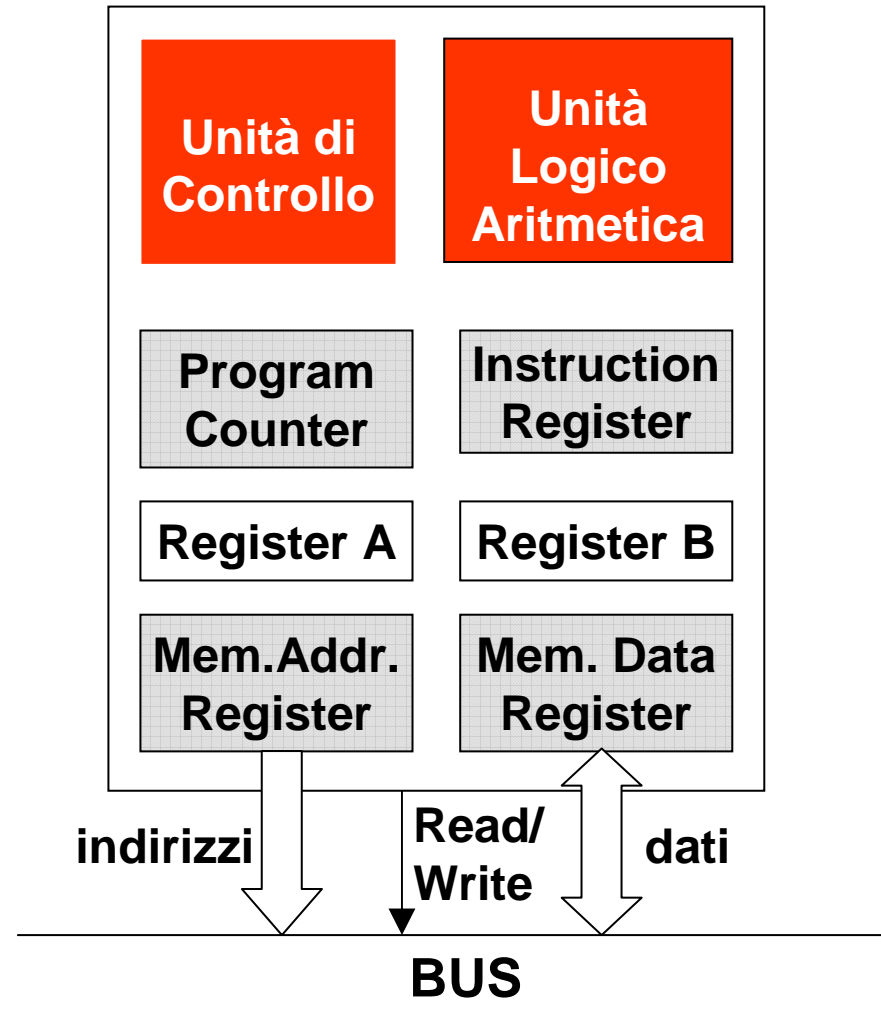
- Program Counter (PC)
- Mem. Address Reg. (MAR)
- Mem. Data Register (MDR)
- Instruction Register (IR)

I registri speciali non sono accessibili dalle istruzioni

Connessione della CPU con il sistema

I vari componenti interni della CPU sono comunicanti tramite connessioni interne.

La CPU è connessa al resto del sistema tramite il BUS (linee indirizzi, dati e controllo).



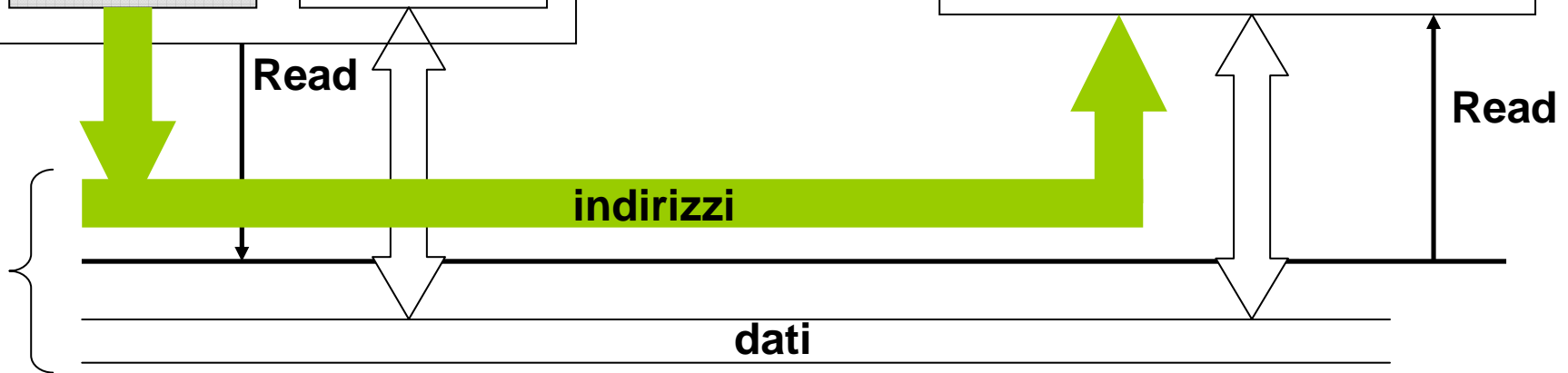
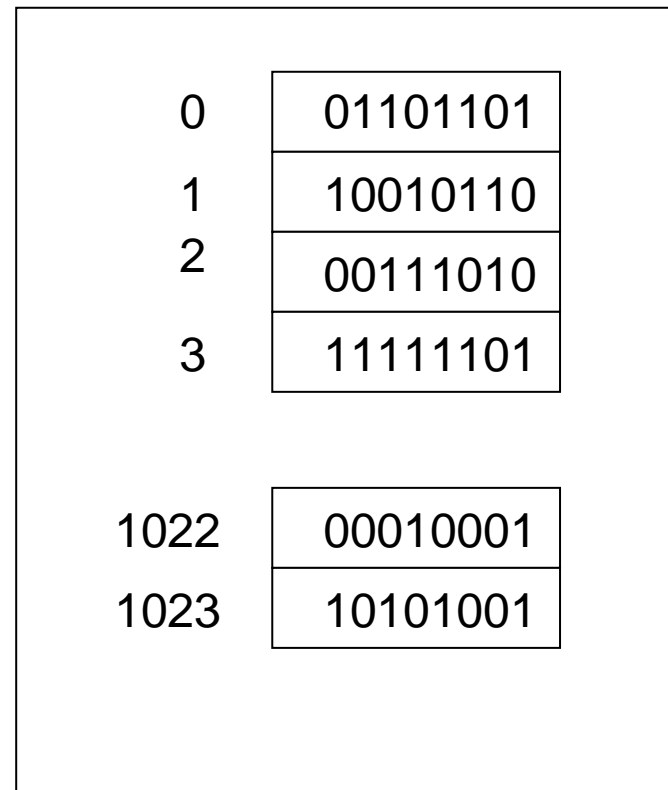
Trasferimento CPU-memoria

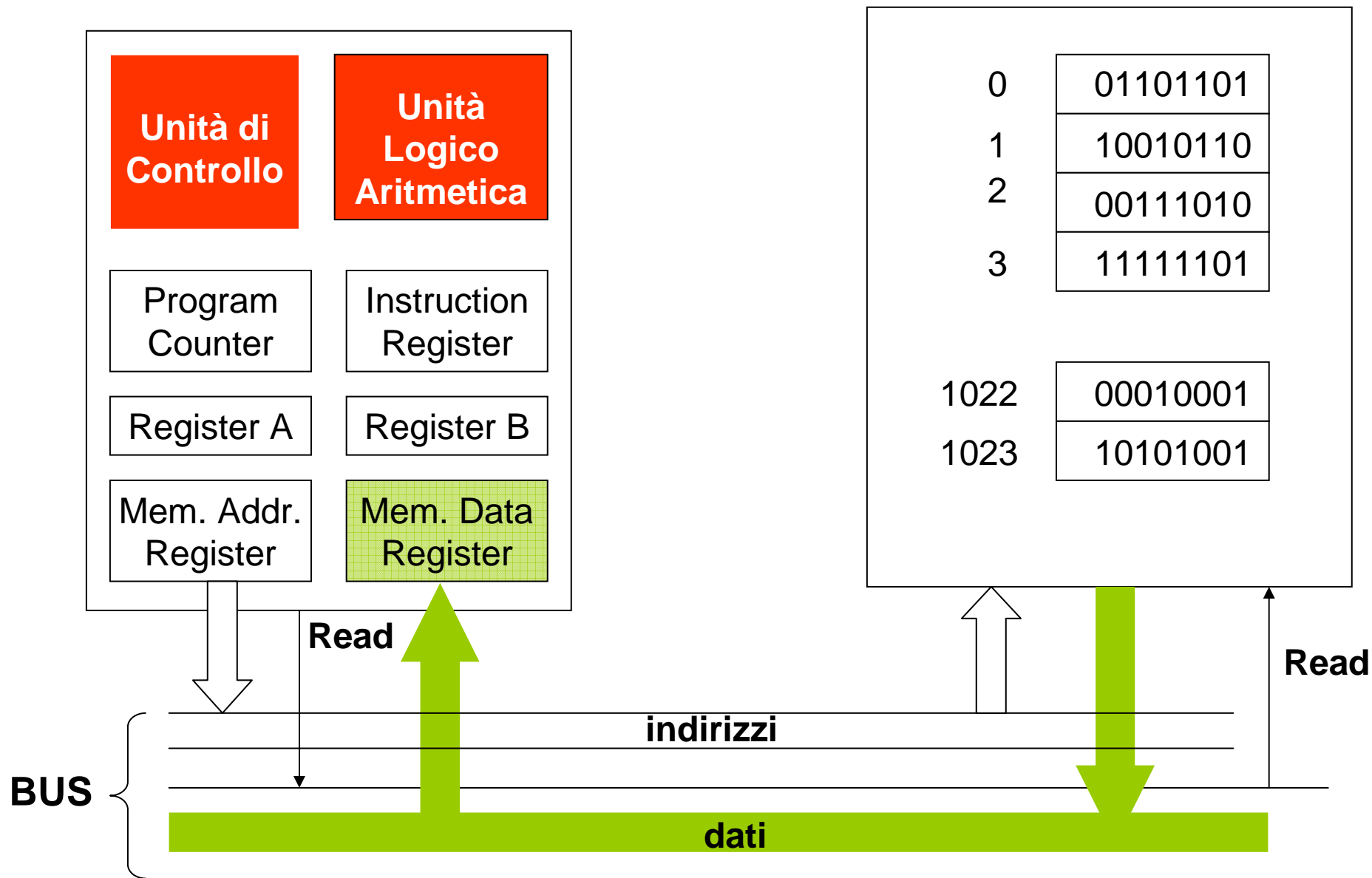
Qualunque sia il trasferimento da realizzare, la CPU (master) deve precisare l'indirizzo del dato da trasferire.

In queste operazioni, la memoria è comunque uno slave e “subisce” l'iniziativa della CPU, ricevendo da questa l'indirizzo del dato da trasferire e l'informazione sull'operazione da realizzare (lettura o scrittura)

Trasferimento memoria → CPU (lettura)

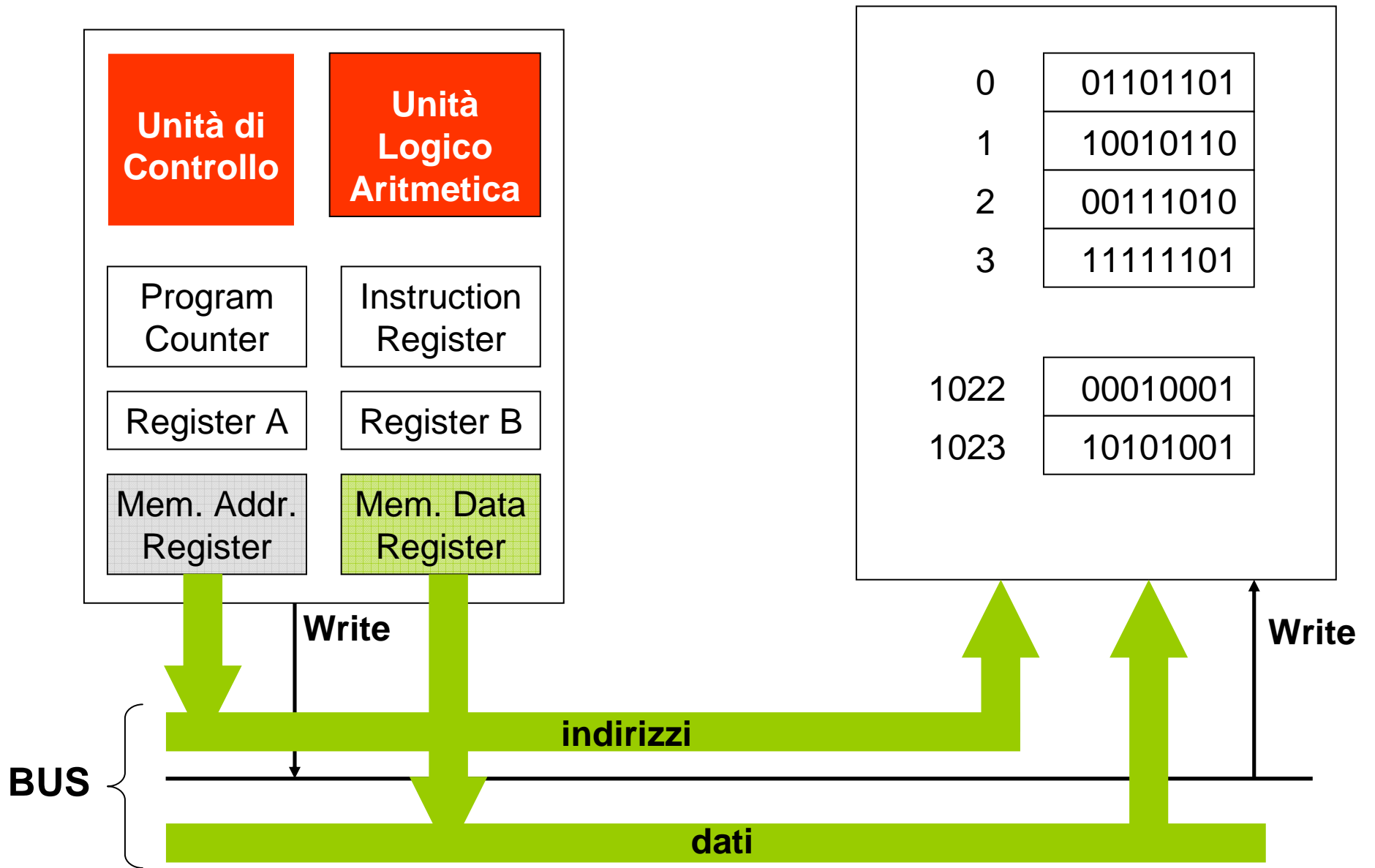
- 1) la CPU scrive l'indirizzo del dato da trasferire sul MAR che lo propagherà alle linee indirizzi del bus. Contemporaneamente, segnala sulle linee di controllo che si tratta di una lettura.
- 2) la memoria riceve, tramite il bus, l'indirizzo e l'indicazione dell'operazione da effettuare. Copia il dato dal registro individuato sulle linee dati del bus.
- 3) il dato richiesto, tramite le linee dati del bus, arriva al MDR della CPU. Da qui sarà spostato verso gli altri registi interni.

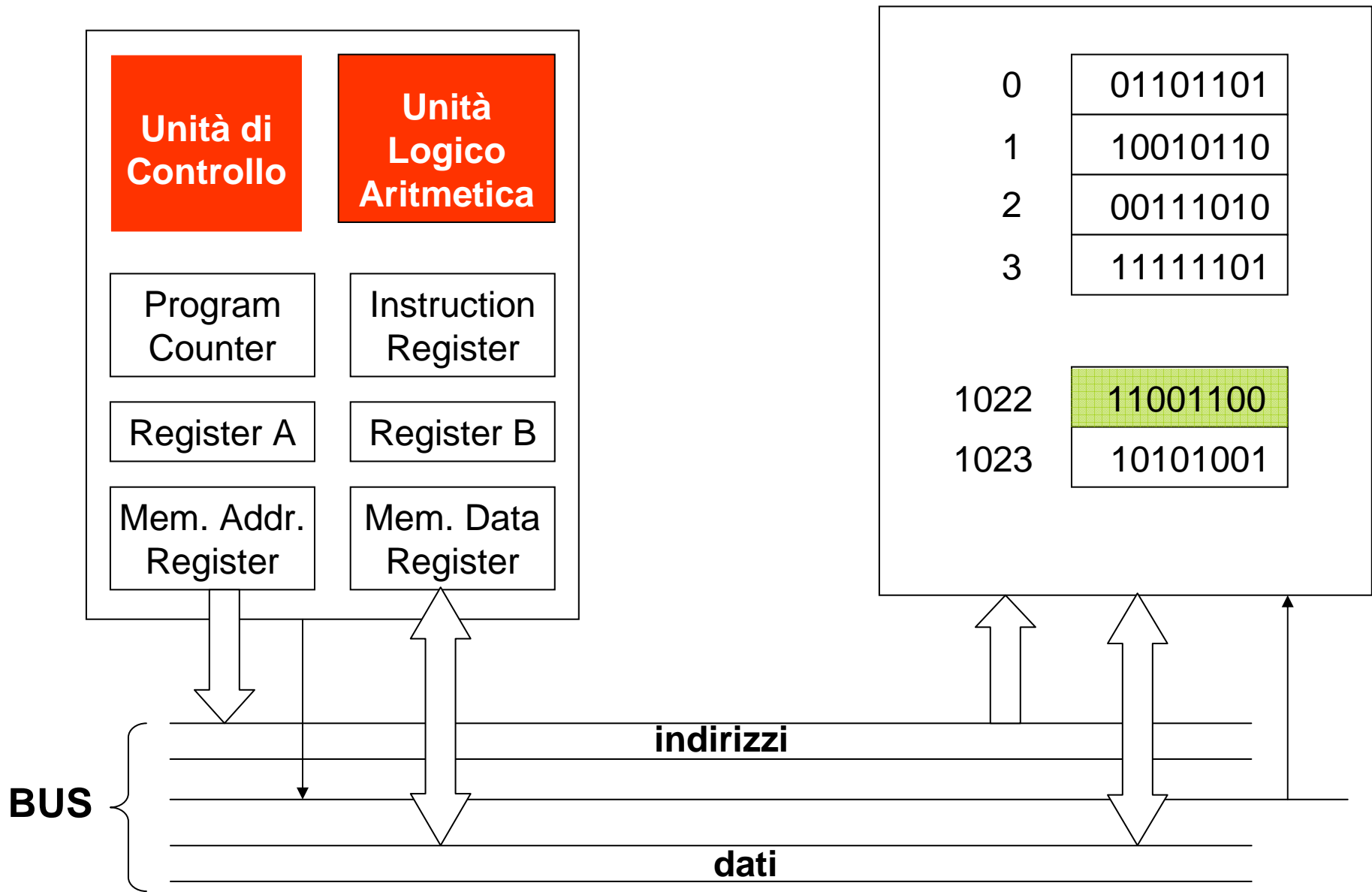




Trasferimento CPU → memoria (scrittura)

- 1) la CPU scrive l'indirizzo del dato da trasferire sul MAR, mentre il dato viene copiato sul MDR. Il contenuto dei due registri viene propagato sulle linee indirizzi e dati del bus. Contemporaneamente, la CPU segnala sulle linee di controllo che si tratta di una scrittura.
- 2) la memoria riceve, tramite il bus, l'indirizzo, il dato e l'indicazione dell'operazione da effettuare. Copia il dato dalle linee dati del bus al registro individuato dall'indirizzo.





Esempio di esecuzione di una istruzione

Consideriamo un'istruzione del tipo:

ADD (1021),(1022),1023

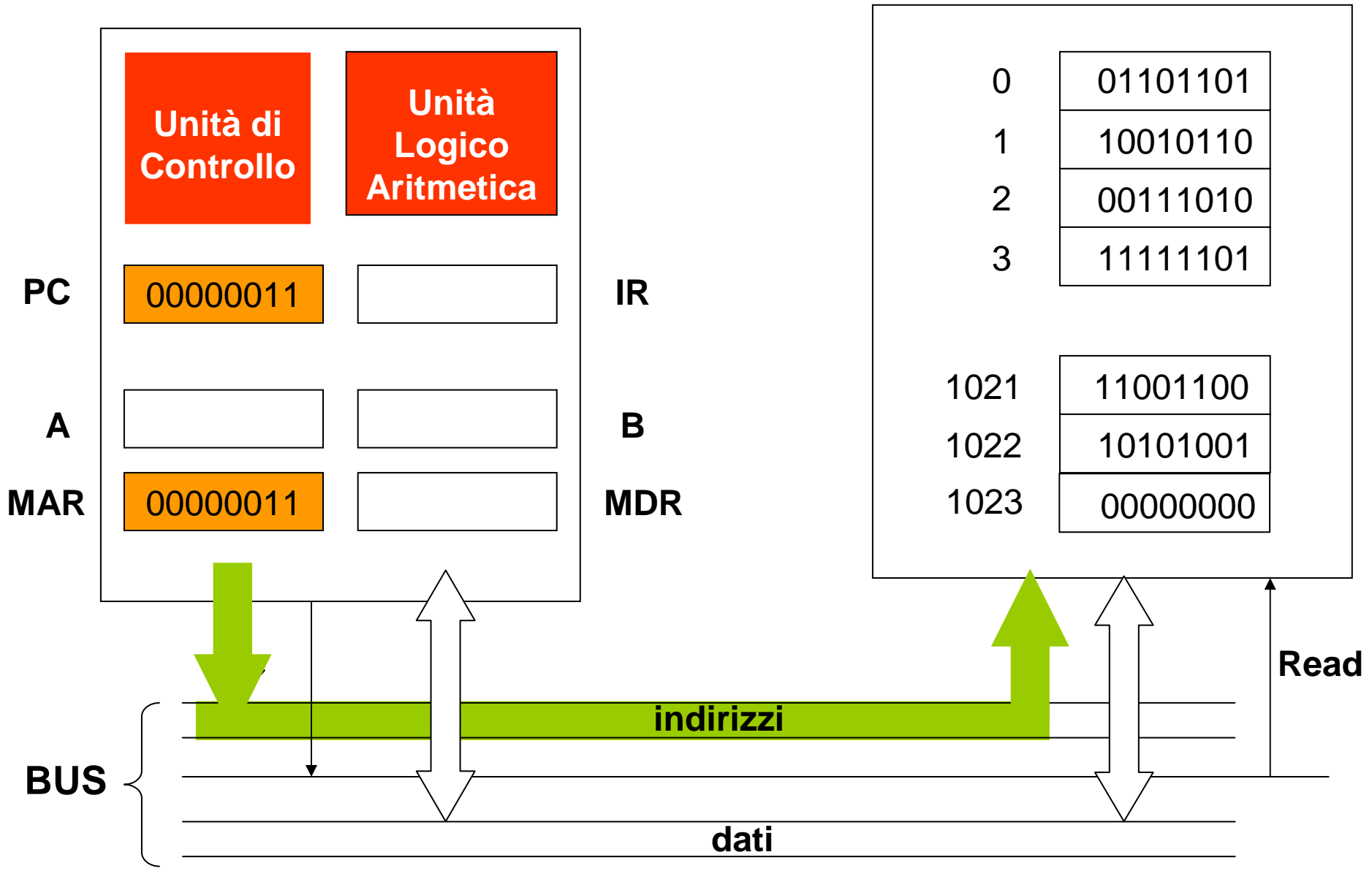
Il cui significato è:

“somma i valori che trovi nei registri di memoria di indirizzo 1021 e di indirizzo 1022 e memorizza il risultato nel registro di indirizzo 1023”.

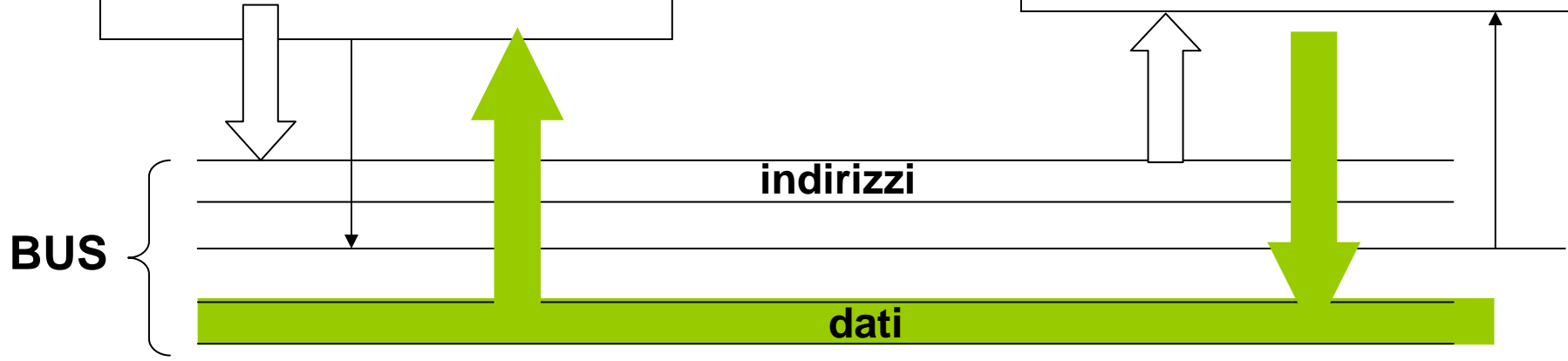
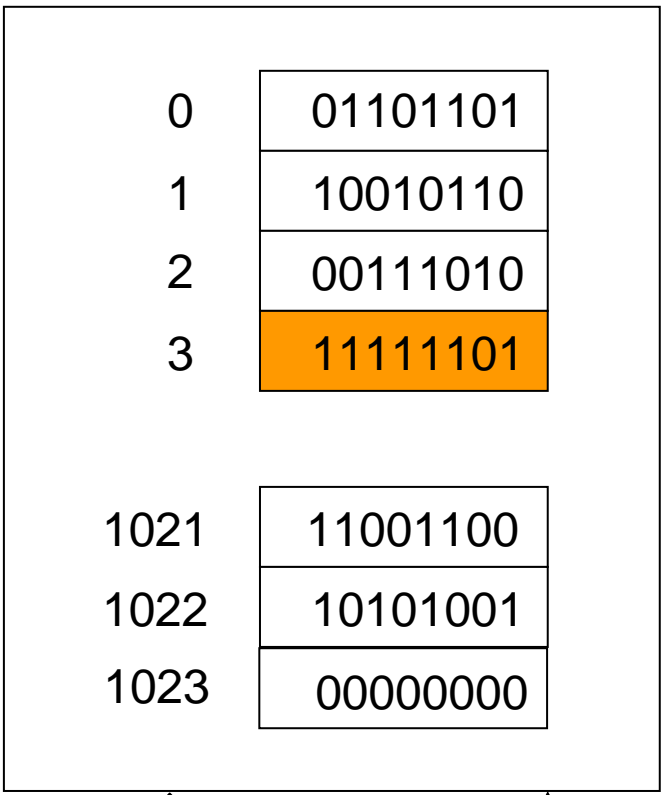
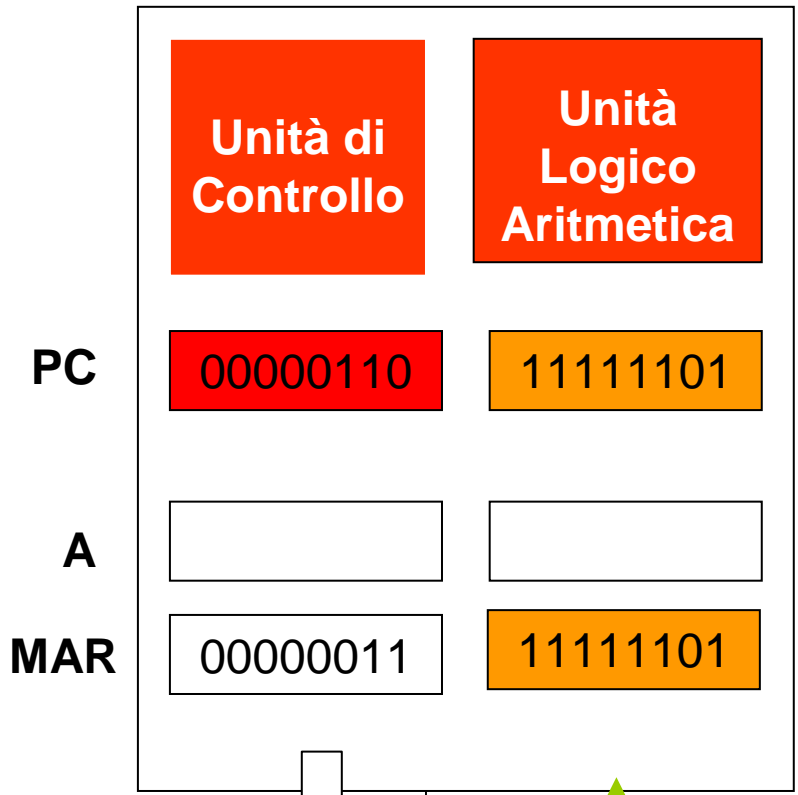
Supponiamo inoltre che l'istruzione si trovi memorizzata nel registro di memoria di indirizzo 3.

Consideriamo le varie fasi necessarie per l'esecuzione di questa istruzione...

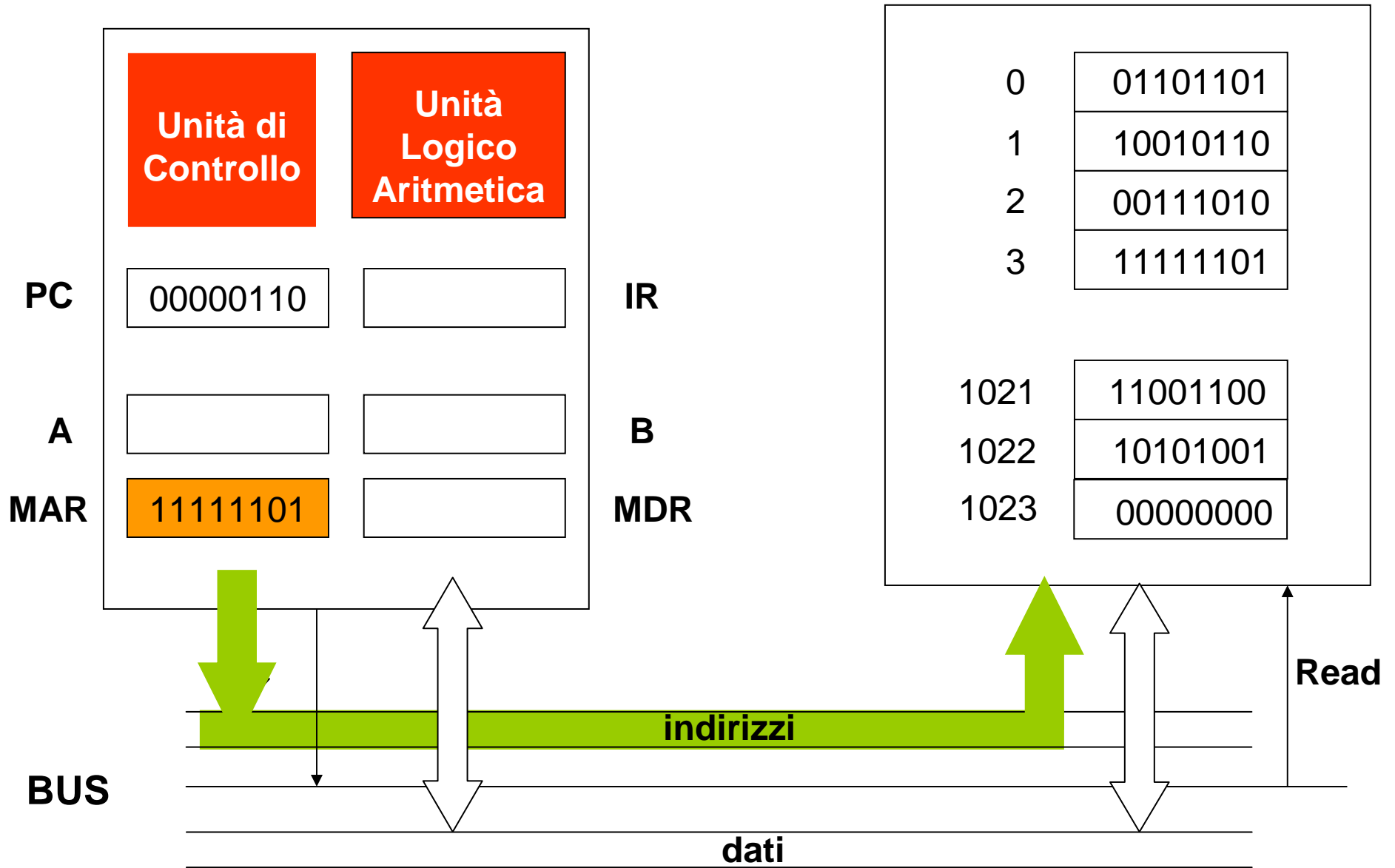
Fase FETCH



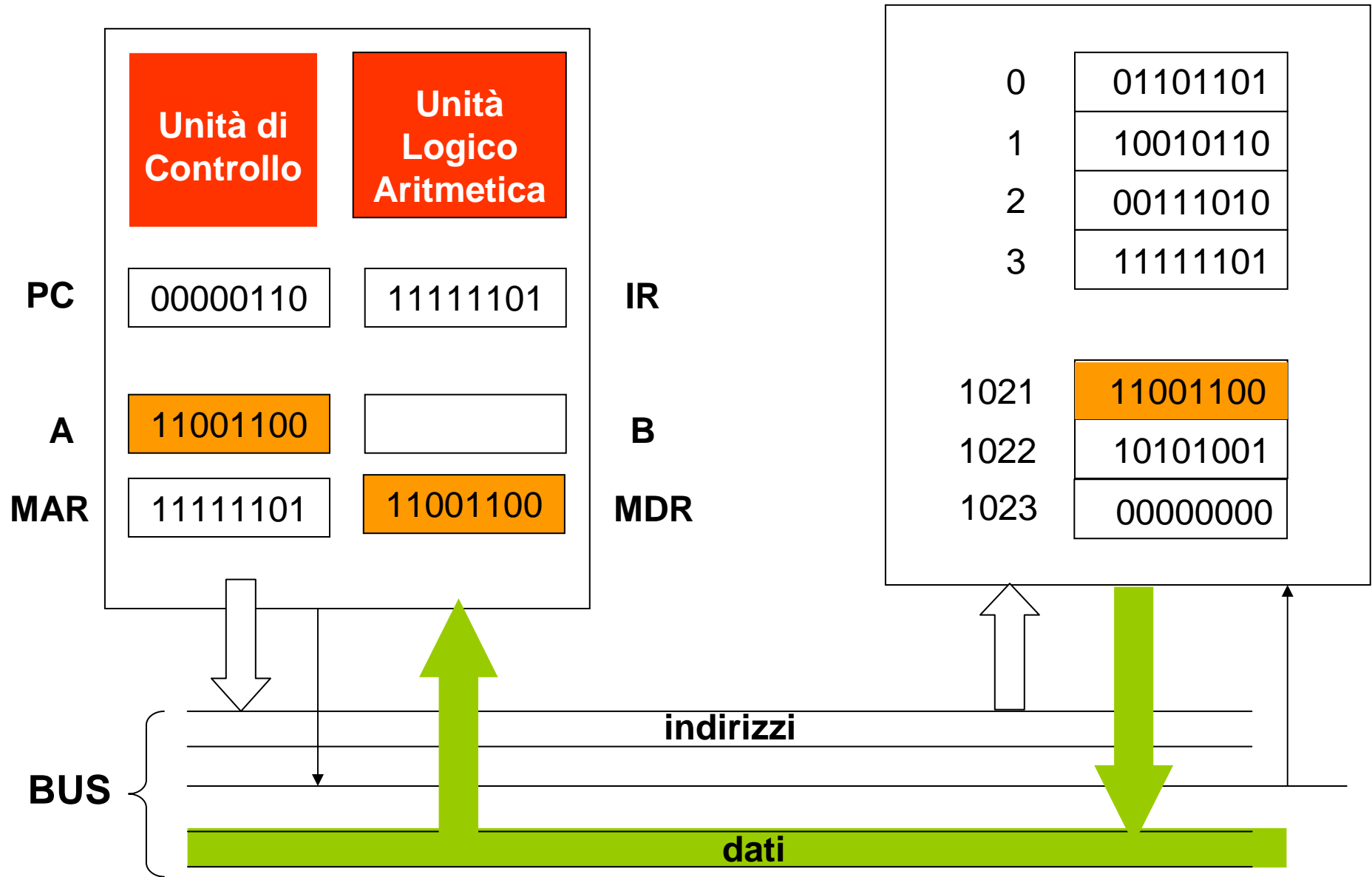
Fase FETCH



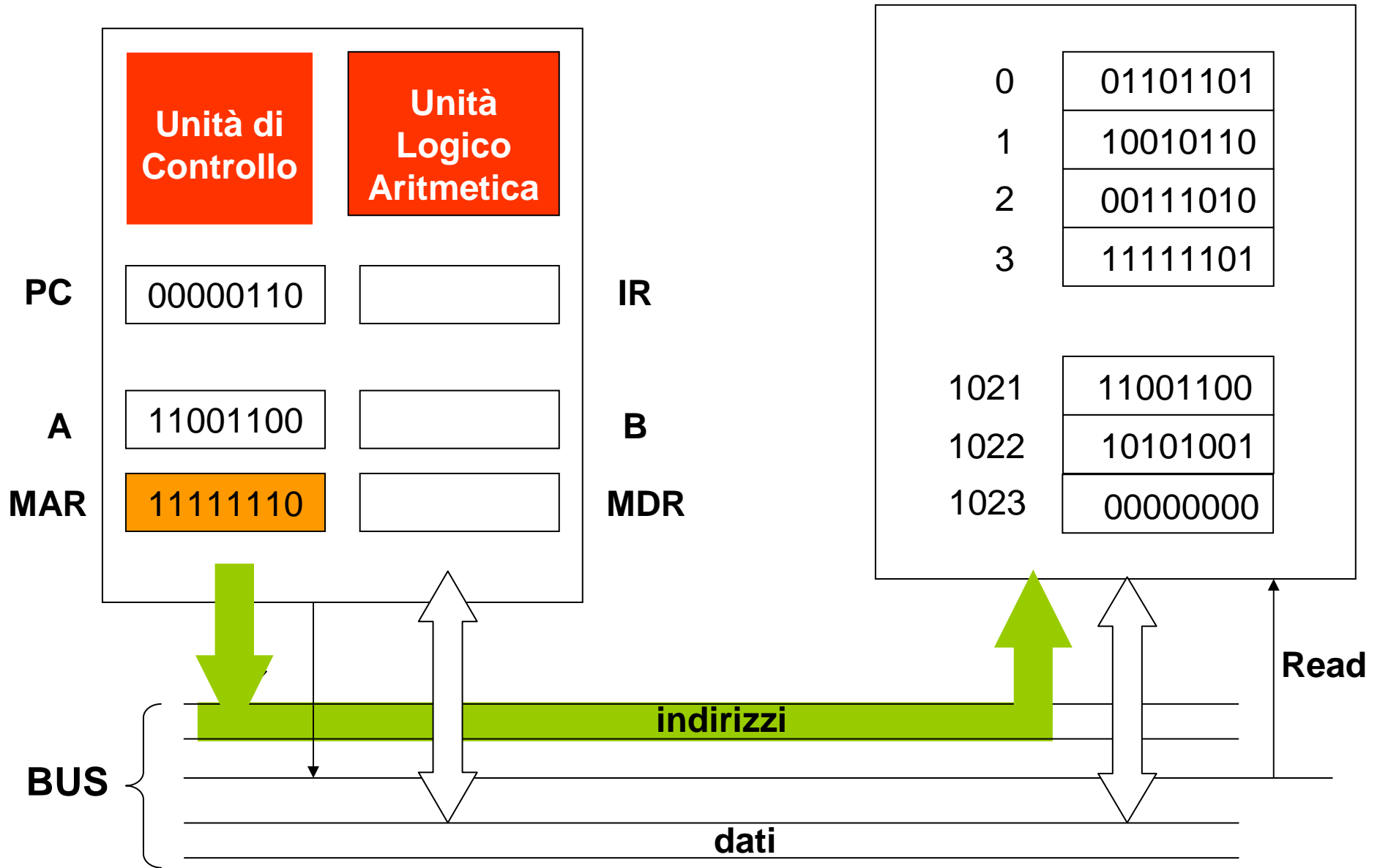
Fase OPERAND ASSEMBLY



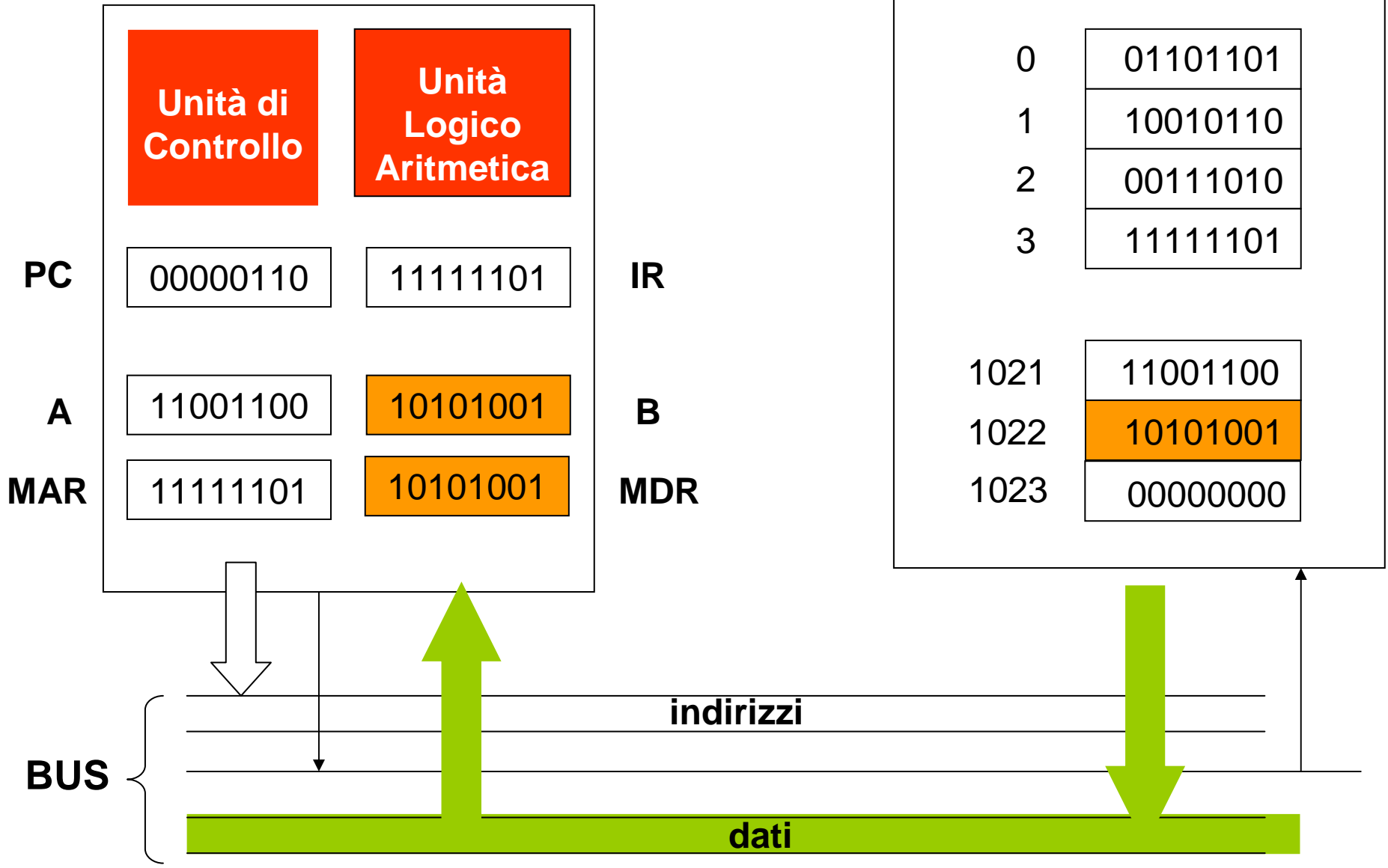
Fase OPERAND ASSEMBLY



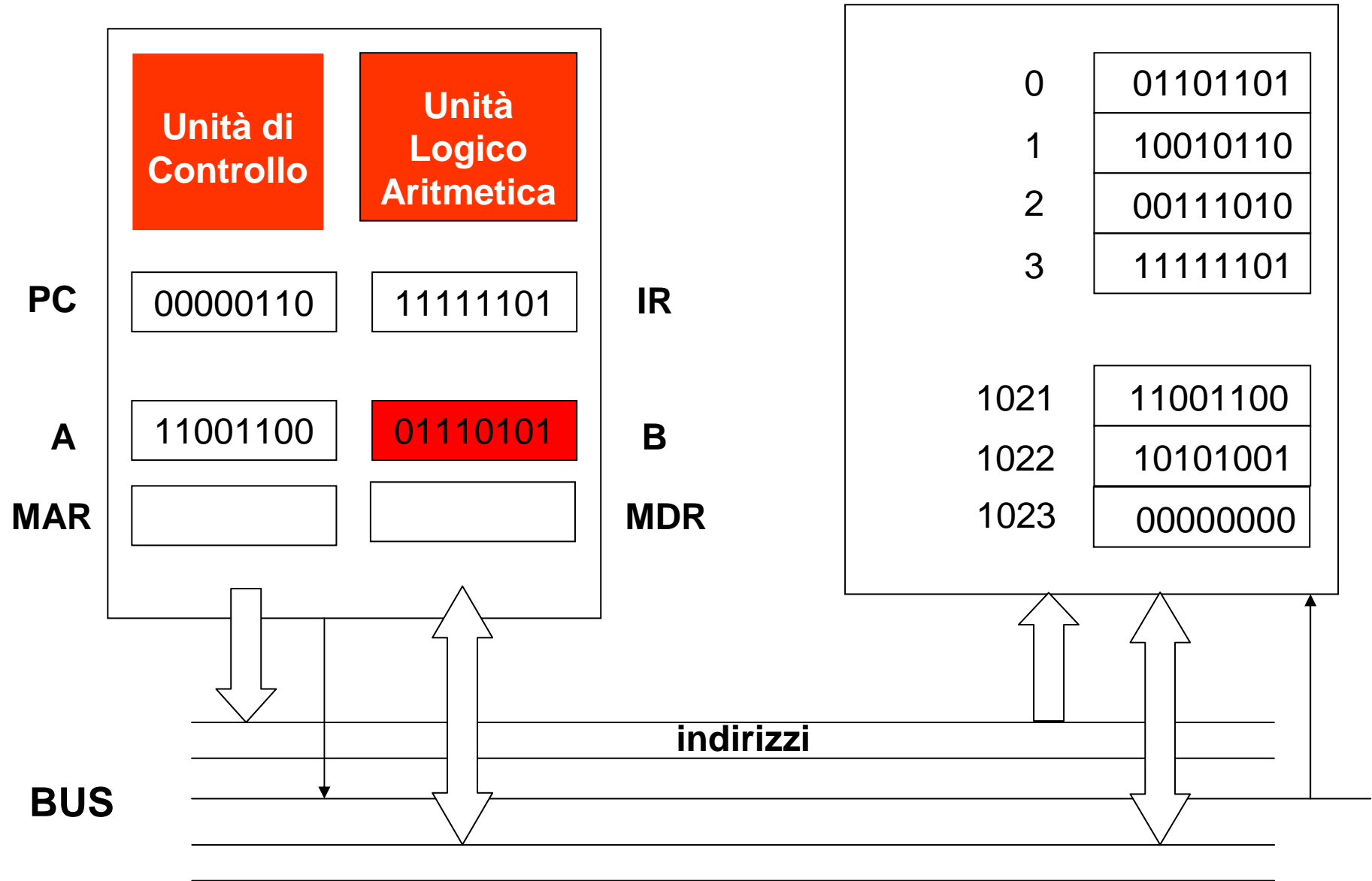
Fase OPERAND ASSEMBLY



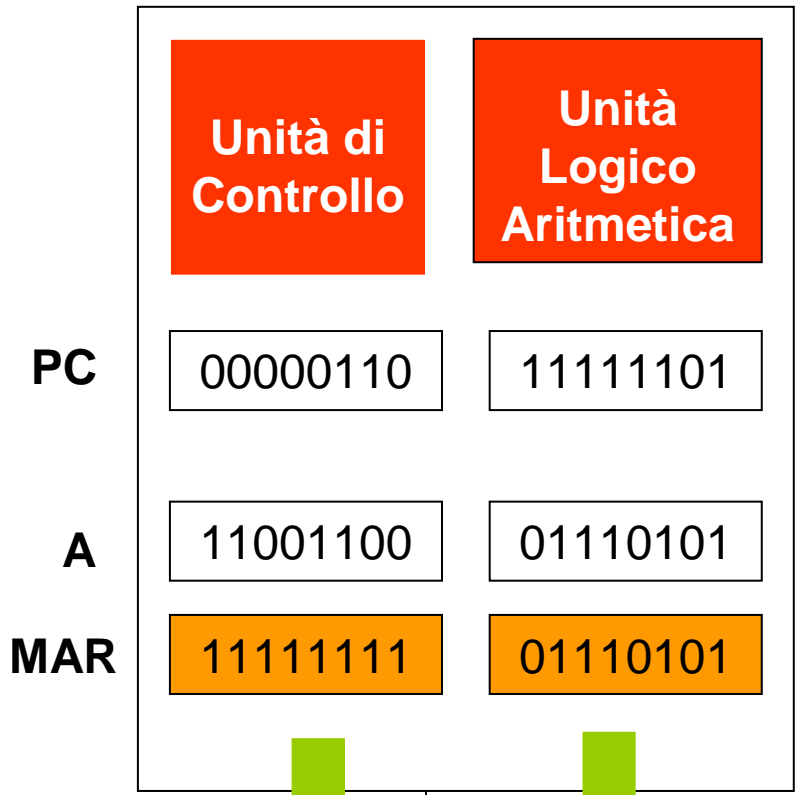
Fase OPERAND ASSEMBLY



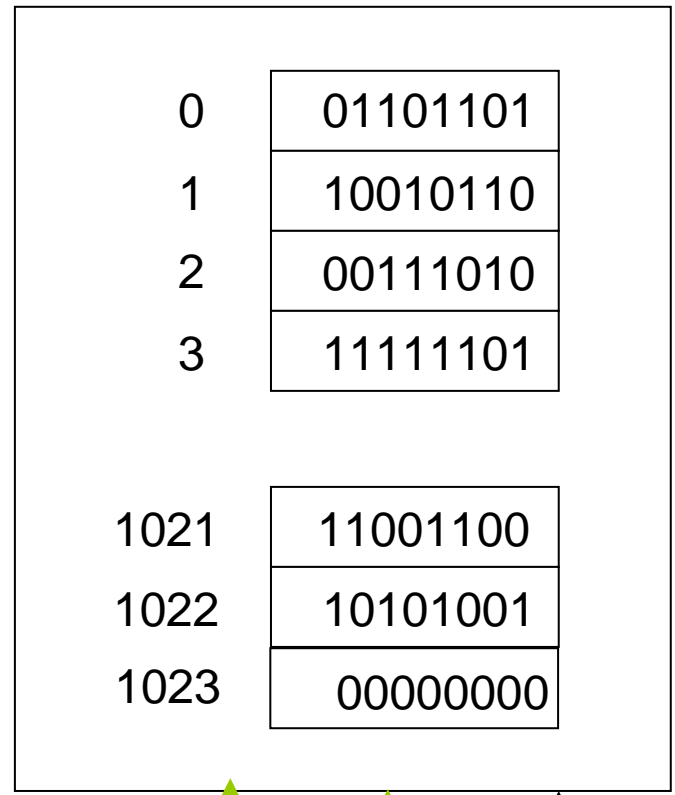
Fase EXECUTE



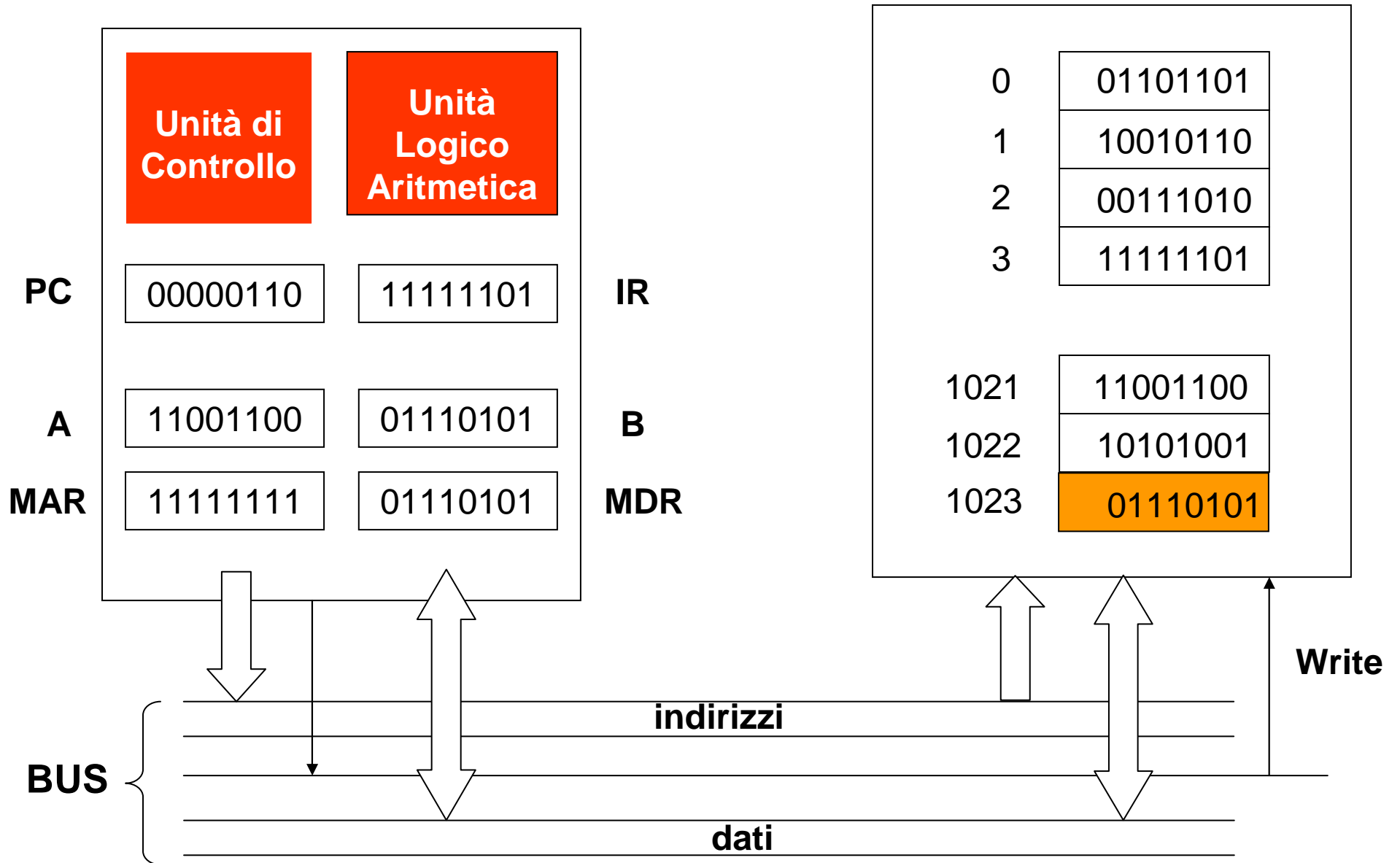
Fase STORE



IR
B
MDR

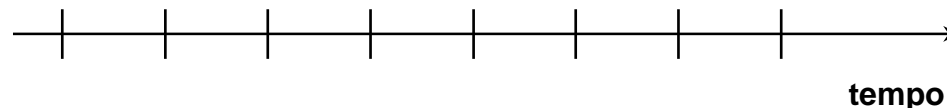


Fase STORE



Clock

- La CPU è sincronizzata da un orologio interno che procede a velocità costante (clock)
- I “clock ticks” definiscono gli istanti possibili per la progressione dei singoli passi eseguiti dal processore:



- tempo di ciclo= intervallo tra due ticks = secondi per ciclo
- clock rate (frequenza) = cicli al secondo (1 Hz. = 1 ciclo/sec)

1 MegaHertz= 1MHz = 10^6 cicli/sec

1 GigaHertz = 1GHz = 10^9 cicli/sec



Frequenze di clock maggiori indicano CPU più veloci